

# CS314 Assignment 2: Object-Oriented Programming Exercise

---

**Due Date – Class Model & Critique: Tuesday, September 18, 2012, before midnight**

**Due Date – Program & Problem Report : Tuesday, October 2, 2012, before Midnight.**

**Submit via RamCT**

**This is a pair-design/programming assignment. You are required to work with a partner on this assignment.**

## **Assignment Objective**

The purpose of this assignment is to give you working experience related to making significant changes to code written by team. A secondary objective is to help you develop your program modeling skills by requiring you to develop a class model of the program you will be revising.

## **Summary of Deliverables**

Here are the specific items you are required to submit. Details are given in the next Assignment Activities section

1. **Class model and critique of program static structure:** Submit as a single PDF file on September 18, 2012, before midnight.
2. **Modified program:** Submit as a jar file on October 2, 2012, before midnight.
3. **Report on the ease or difficulty of modifying the assigned program:** Submit as a PDF file on October 2, 2012, before midnight.

## **Assignment Activities**

In this assignment you will be assigned a program and will be required to do the following:

1. Produce a class model of the program design and use the model to write a critique (between 2 to 5 pages using 12 point font) of the design. In your critique comment on the following:

- a. Identify classes that are not cohesive
- b. Identify cases of bad distribution of responsibilities across the classes
- c. Identify bad uses of inheritance
- d. Identify missed opportunities for applying polymorphism

In all of the above, you must justify your criticisms. **You must submit this via RamCT before midnight Tuesday, September 18, 2012.**

2. Extend/Modify the program with the following features (**Due October 2, 2012**, before midnight):
  - a. **Create the initial airport system using information contained in an input file:**  
Information about airports, airlines, flights and their associated information will now be read from a file. When the program first starts it will read the input file and create the initial airport system using the information in the file. The format of the file is given in the section AMS File Format.
  - b. **Store information about the current airport system in a file:** The program should be able to store information about the airport system in a file when requested. This feature is similar to the display airport system feature: instead of outputting the information on the screen the information is sent to a file to be stored. The format of this file will be the same as the format of the input airport system file.
  - c. **Create flight sections with a layout:** In the previous assignment a flight section was created using only the number of seats and columns. In this assignment you will be required to create sections given a layout that identifies window and aisle seats. See the AMS file format section for an example of how the layouts are to be specified.
  - d. **Associate one-way prices for seats on an airline for flights between an origin and destination:** Given an airline, the pricing for all seats in a particular flight class for all airline flights between an origin and destination is the same. For example, the price of an economy seat on any American Airways flight from Denver to Seattle is \$300, while an economy seat on any USAircorp flight from Denver to Seattle is \$200.
  - e. **Book a flight using a seating preference:**
  - f. **Provide a simple intuitive text-based user interface for the system:** The interface will allow a human user to do the following
    - i. Create an airport system by using information provided in an input file.
    - ii. Change the price associated with seats in a flight section (all seats in a flight section have the same price).

- iii. Query the system for flights with available seats in a given class (e.g., economy, business) that leave from a specified airport and arrive at specified airport on a particular date. The query operation should list all the available flights found and its prices.
  - iv. Change the seat class (e.g., economy) pricing for an origin and destination for a given airline.
  - v. Book a seat given a specific seat on a flight.
  - vi. Book a seat on a flight given only a seating preference: The program should allow a user to book a seat on a particular flight using only a seating preference and a flight class. There will only be two seating preferences: Window and Aisle. This booking service will look for an available seat in the flight section with the seating preference. If one is found then the seat is booked. If one is not found, then the system will book any available seat in the specified section, if any.
  - vii. Display details of the airport system.
  - viii. Store information about the airport system in a specified file.
3. Write a report on the difficulties you encountered making the above changes to the code. NOTE: You are NOT allowed to reuse your code in this assignment. We will check this program against your original program using a program we use to check similarities between programs.

## AMS File Format

AMS information will be stored in a file using the following format:

AMS ::= [list-of-airport-codes] {list-of-airlines}

list-of-airport-codes ::= comma-separated strings

list-of-airlines ::= airline-name1[flightinfo-list1], airline-name2[flightinfo-list2], airline-name3[flightinfo-list3], ...

flightinfo-list ::= flightID1|flightdate1|originAirportCode1|destinationAirportCode1[flightsection-list1], flightID2|flightdate2|originAirportCode2|destinationAirportCode2[flightsection-list2], ...

flightdate ::= year, month, day-of-month, hours, minutes

flightsection-list ::= sectionclass: seat-price: layout: number-of-rows, ...

sectionclass ::= F, B, E (for first class, business class, economy class)

layout ::= S, M, W (where S is a seat layout with 3 columns with an aisle between columns 1 and 2, M is a seat layout with 4 columns with an aisle between columns 2 and 3, and W is a seat layout with 10 columns with aisles between columns 3 and 4, and between columns 7 and 8)

Example: An airport system with 4 airports, 4 airlines - AMER (in red), UNTD (in green), FRONT, USAIR - and associated flights.

```
[DEN, NYC, SEA, LAX]{AMER[AA1|2011, 10, 8, 16, 30|DEN|LAX[E:200:S:4,F:500:S:2],  
AA2|2011, 8, 9, 7, 30|LAX|DEN[E:200:S:5,F:500:S:3], ...], UNTD[UA21|2011, 11, 8, 12,  
30|NYC|SEA[E:300:S:6, F:800:S:3], UA12|2011, 8, 9, 7, 30|SEA|DEN[B:700:S:5, F:1200:S:2],  
...], FRONT[...], USAIR[...]}
```

## Grading

Points will be allocated as follows:

- Assignment 2: 100
  - Class model: 15 points
  - Modifications: 80 points (the points will be tempered by you report on difficulties; you can still get the full 60 points even if not all features are fully implemented, if you report, with good evidence, on how the bad design hindered your progress. **You will be required to meet with me or Sai Mandalaparty at least 5 days before the deadline if you have difficulties making the modifications because of a bad design.**)
  - Report on difficulties: 5 points

## Important Notes

**As for Assignment 1, problem statement may be incomplete:** Do not make assumptions about what is needed! If the information is not provided, ask questions.

**Please start early:** As soon as you get your assigned program please start to work on it. **Note that the class diagram/model is due September 18, 2012.**

**Please make initial contact with me and Sai at least 5 days before the October 2, 2012 deadline if you encounter difficulties with the design of the program you are assigned to. This requires that you start the assignment early.**